

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1 1. (Currently amended) A computer-implemented method of piggybacking
2 a message invalidating cached data on a response to a data request, the method
3 comprising:

4 in a hierarchy of caches, passing a data request toward a data server;
5 at an upstream cache, detecting the invalidation of a set of data cached on
6 one or more downstream caches, including a first downstream cache, wherein the
7 set of data includes no data requested in the data request;
8 in a response to the data request, including an invalidation message to
9 invalidate the set of data attached to with the response to the first downstream
10 cache to invalidate the set of data; and
11 forwarding the response including the invalidation message toward the
12 first downstream cache.

1 2. (Original) The method of claim 1, further comprising:
2 adding the set of data to a list of data to be invalidated on downstream
3 caches.

1 3. (Original) The method of claim 2, further comprising:
2 removing the set of data from the list of data after data request responses
3 including messages to invalidate the set of data have been forwarded to each of
4 the one or more downstream caches.

1 4. (Original) The method of claim 1, wherein the set of data includes data
2 requested in the data request.

1 5. (Cancelled)

1 6. (Original) The method of claim 1, further comprising, in the first
2 downstream cache:

3 receiving the response;

4 retrieving the message;

5 invalidating the set of data if the set of data is cached on the first
6 downstream cache; and

7 forwarding the response toward a second downstream cache.

1 7. (Original) The method of claim 1, further comprising, in the first
2 downstream cache:

3 receiving the response;

4 removing the message from the response;

5 invalidating the set of data if the set of data is cached on the first
6 downstream cache; and

7 serving the response to a client that initiated the data request.

1 8. (Original) The method of claim 1, wherein said detecting comprises:
2 receiving an invalidation message originated by the data server.

1 9. (Original) The method of claim 1, wherein said detecting comprises:
2 receiving a manual notification of the invalidation of the set of data.

1 10. (Original) The method of claim 1, wherein the data request is initiated

2 by a first requestor and the response is targeted to a different requestor.

1 11. (Original) The method of claim 1, wherein the upstream cache is a
2 cache local to the data server.

1 12. (Original) The method of claim 1, wherein the upstream cache and the
2 first downstream cache are members of a cache cluster.

1 13. (Currently amended) A computer readable medium storing instructions
2 that, when executed by a computer, cause the computer to perform a method of
3 piggybacking a message invalidating cached data on a response to a data request,
4 the method comprising:

5 in a hierarchy of caches, passing a data request toward a data server;
6 at an upstream cache, detecting the invalidation of a set of data cached on
7 one or more downstream caches, including a first downstream cache, wherein the
8 set of data includes no data requested in the data request;

9 in a response to the data request, including an invalidation message to
10 invalidate the set of data attached to with the response to the first downstream
11 ~~cache to invalidate the set of data;~~ and

12 forwarding the response including the invalidation message toward the
13 first downstream cache.

1 14. (Currently amended) An automated method of asynchronously
2 communicating a side effect of a first data request in a response to a second data
3 request, the method comprising:

4 in a computing environment comprising a data server and a plurality of
5 caches, processing a first data request to produce a first response;
6 identifying a side effect of the first data request;

7 communicating the side effect to a first cache upstream of one or more
8 downstream caches;
9 at the first cache:
10 identifying a second response to a second data request;
11 including notification of the side effect in the second response, wherein the
12 side effect is unrelated to the second response; and
13 forwarding the second response including the notification of the side effect
14 toward a first downstream cache; and
15 at the first downstream cache, applying the side effect.

1 15. (Original) The method of claim 14, wherein the side effect comprises
2 invalidation of data cached on the first downstream cache.

1 16. (Original) The method of claim 14, wherein the side effect comprises
2 propagation of cache configuration data.

1 17. (Original) The method of claim 14, wherein the side effect comprises a
2 password.

1 18. (Original) The method of claim 14, wherein the side effect comprises
2 an update to a cache program.

1 19. (Original) The method of claim 14, wherein the side effect comprises a
2 replacement cache program.

1 20. (Original) The method of claim 14, wherein said applying the side
2 effect at the first downstream cache comprises applying the side effect after
3 forwarding the second response.

1 21. (Original) The method of claim 20, wherein said applying the side
2 effect at the first downstream cache comprises applying the side effect before a
3 specified event.

1 22. (Original) The method of claim 14, further comprising:
2 at the first cache, tracking which of the one or more downstream caches
3 has been notified of the side effect;
4 wherein a downstream cache other than the first downstream cache
5 receives notification of the side effect in a communication other than the second
6 response.

1 23. (Currently amended) A computer readable medium storing instructions
2 that, when executed by a computer, cause the computer to perform a method of
3 asynchronously communicating a side effect of a first data request in a response to
4 a second data request, the method comprising:
5 in a computing environment comprising a data server and a plurality of
6 caches, processing a first data request to produce a first response;
7 identifying a side effect of the first data request;
8 communicating the side effect to a first cache upstream of one or more
9 downstream caches;
10 at the first cache:
11 identifying a second response to a second data request;
12 including notification of the side effect in the second response, wherein the
13 side effect is unrelated to the second response; and
14 forwarding the second response including the notification of the side effect
15 toward a first downstream cache; and
16 at the first downstream cache, applying the side effect.

1 24. (Currently amended) A system for piggybacking notification of a side
2 effect of a first data request in a response to a second data request, comprising:
3 a data server configured to serve data in response to data requests;
4 one or more downstream caches configured to cache the served data; and
5 an upstream cache logically located between the data server and the one or
6 more downstream caches, wherein the upstream cache is configured to include, in
7 a response to one data request from the downstream cache, notification of a first
8 side effect of a different data request, wherein the side effect is unrelated to the
9 response to one data request.

1 25. (Original) The system of claim 24, wherein the upstream cache
2 comprises:

3 a subscription table identifying each of the one or more downstream
4 caches; and
5 a list of side effects that the one or more downstream caches are to be
6 notified of.

1 26. (Original) The system of claim 25, wherein the upstream cache is
2 further configured to remove the first side effect from the list of side effects after
3 each of the one or more downstream caches have been notified of the first side
4 effect.

1 27. (Original) The system of claim 24, wherein the first side effect
2 comprises the invalidation of a set of data.

1 28. (Original) The system of claim 24, wherein:
2 the one or more downstream caches include a final downstream cache
3 coupled to a client that initiated the one data request; and

4 the final downstream cache is configured to remove the notification of the
5 first side effect of the different data request from the response before serving the
6 response to the client.

1 29. (Original) The system of claim 24, wherein the upstream cache is a
2 local cache of the data server.